

CLAIMS

What is claimed is:

- 1 1. A method of software loading and initialization in a distributed network of nodes, the
2 method comprising the computer-implemented steps of:
3 providing a master node;
4 providing software package storage means on said master node for storing software
5 packages and boot images that the nodes in the network will be using as well
6 as older versions that are kept for regressing a node back to a previous boot
7 image or software package version;
8 providing node information storage means on said master node for storing preferred
9 software version information, node type, and other pertinent information for
10 each node in the network;
11 wherein a node performing an initial boot requests a boot image and software
12 packages from said master node;
13 retrieving said node's preferred software version information from said node
14 information storage means;
15 extracting boot image and software packages from said software package storage
16 means using said node's preferred software version information;
17 delivering the extracted boot image and software packages to said node;
18 wherein said node stores the extracted boot image and software packages in its local
19 persistent storage and wherein software version information is extracted from
20 the software packages and stored in the local persistent storage; and
21 wherein said node reboots and executes the boot image stored in the local persistent
22 storage.

1 2. A method as recited in Claim 1, wherein said node, based on a command from said
2 master node, does not store the software packages in the local persistent storage device,
3 allowing said master node to download test software packages to said node and temporarily
4 run said node using the test software packages, and wherein when said node reboots, the test
5 software packages will no longer exist on said node.

1 3. A method as recited in Claim 1, wherein said retrieving step creates said node's
2 preferred software version information from said node information storage means based on
3 functional features requested by said node.

1 4. A method as recited in Claim 1, wherein said node verifies the software version
2 information with said master node.

1 5. A method as recited in Claim 4, wherein if said node has the correct software
2 versions, then said node completes booting by executing the software packages stored in the
3 local persistent storage.

1 6. A method as recited in Claim 4, wherein if said node does not have the correct
2 software versions, said master node retrieves correct software packages from said software
3 package storage means and sends the correct software packages to said node, and wherein
4 said node stores the correct software packages in the local persistent storage and completes
5 booting by executing the correct software packages stored in the local persistent storage.

1 7. A method as recited in Claim 1, wherein the master node has the ability to categorize
2 nodes into classes where all of the nodes in a particular class of nodes have the same
3 software configuration and may have differing processor types.

1 8. A method as recited in Claim 1, wherein a software package contains version
2 information, dependency information, and other metadata information pertaining to software
3 in the package.

1 9. A method as recited in Claim 1, wherein a boot image is customized for a particular
2 type of node and provides basic low-level communications.

1 10. A method of software loading and initialization in a distributed network of nodes, the
2 method comprising the computer-implemented steps of:

3 providing a master node;

4 providing software package storage means on said master node for storing software
5 packages and boot images that the nodes in the network will be using as well
6 as older versions that are kept for regressing a node back to a previous boot
7 image or software package version;

8 providing node information storage means on said master node for storing preferred
9 software version information, node type, and other pertinent information for
10 each node in the network;

11 wherein a node performing an initial boot requests a boot image and software
12 packages from said master node;

13 retrieving said node's preferred software version information from said node
14 information storage means;

15 extracting boot image and software packages from said software package storage
16 means using said node's preferred software version information; and

17 delivering the extracted boot image and software packages to said node.

1 11. A method as recited in Claim 10, wherein said node stores the extracted boot image
2 and software packages in its local persistent storage and wherein software version
3 information is extracted from the software packages and stored in the local persistent storage.

1 12. A method as recited in Claim 11, wherein said node, based on a command from said
2 master node, does not store the software packages in the local persistent storage device,
3 allowing said master node to download test software packages to said node and temporarily
4 run said node using the test software packages, and wherein when said node reboots, the test
5 software packages will no longer exist on said node.

1 13. A method as recited in Claim 11, wherein said node reboots and executes the boot
2 image stored in the local persistent storage, and wherein said node verifies the software
3 version information with said master node.

1 14. A method as recited in Claim 13, wherein if said node has the correct software
2 versions, then said node completes booting by executing the software packages stored in the
3 local persistent storage.

1 15. A method as recited in Claim 13, wherein if said node does not have the correct
2 software versions, said master node retrieves correct software packages from said software
3 package storage means and sends the correct software packages to said node, and wherein
4 said node stores the correct software packages in the local persistent storage and completes
5 booting by executing the correct software packages stored in the local persistent storage.

1 16. A method as recited in Claim 10, wherein the master node has the ability to
2 categorize nodes into classes where all of the nodes in a particular class of nodes have the
3 same software configuration and may have differing processor types.

1 17. A method as recited in Claim 10, wherein a software package contains version
2 information, dependency information, and other metadata information pertaining to software
3 in the package.

1 18. A method as recited in Claim 10, wherein a boot image is customized for a particular
2 type of node and provides basic low-level communications.
3

1 19. A method as recited in Claim 10, wherein a user installs a composite image onto said
2 master node which is executed and creates boot images, software packages, and node
3 information, and wherein said master node places the boot images and software packages in
4 said software package storage means and the node information in said node information
5 storage means.

1 20. A method as recited in Claim 10, wherein said retrieving step creates said node's
2 preferred software version information from said node information storage means based on
3 functional features requested by said node.

1 21. A computer-readable medium carrying one or more sequences of instructions for
2 software loading and initialization in a distributed network of nodes, which instructions,
3 when executed by one or more processors, cause the one or more processors to carry out the
4 steps of:

5 providing a master node;

6 providing software package storage means on said master node for storing software
7 packages and boot images that the nodes in the network will be using as well
8 as older versions that are kept for regressing a node back to a previous boot
9 image or software package version;

10 providing node information storage means on said master node for storing preferred
11 software version information, node type, and other pertinent information for
12 each node in the network;

13 wherein a node performing an initial boot requests a boot image and software
14 packages from said master node;

15 retrieving said node's preferred software version information from said node
16 information storage means;

17 extracting boot image and software packages from said software package storage
18 means using said node's preferred software version information; and
19 delivering the extracted boot image and software packages to said node.

1 22. A computer-readable medium as recited in Claim 21, wherein said node stores the
2 extracted boot image and software packages in its local persistent storage and wherein
3 software version information is extracted from the software packages and stored in the local
4 persistent storage.

1 23. A computer-readable medium as recited in Claim 22, wherein said node, based on a
2 command from said master node, does not store the software packages in the local persistent
3 storage device, allowing said master node to download test software packages to said node
4 and temporarily run said node using the test software packages, and wherein when said node
5 reboots, the test software packages will no longer exist on said node.

1 24. A computer-readable medium as recited in Claim 22, wherein said node reboots and
2 executes the boot image stored in the local persistent storage, and wherein said node verifies
3 the software version information with said master node.

1 25. A computer-readable medium as recited in Claim 24, wherein if said node has the
2 correct software versions, then said node completes booting by executing the software
3 packages stored in the local persistent storage.

1 26. A computer-readable medium as recited in Claim 24, wherein if said node does not
2 have the correct software versions, said master node retrieves correct software packages from
3 said software package storage means and sends the correct software packages to said node,
4 and wherein said node stores the correct software packages in the local persistent storage and
5 completes booting by executing the correct software packages stored in the local persistent
6 storage.

1 27. A computer-readable medium as recited in Claim 21, wherein the master node has the
2 ability to categorize nodes into classes where all of the nodes in a particular class of nodes
3 have the same software configuration and may have differing processor types.

1 28. A computer-readable medium as recited in Claim 21, wherein a software package
2 contains version information, dependency information, and other metadata information
3 pertaining to software in the package.

1 29. A computer-readable medium as recited in Claim 21, wherein a boot image is
2 customized for a particular type of node and provides basic low-level communications.
3

1 30. A computer-readable medium as recited in Claim 21, wherein a user installs a
2 composite image onto said master node which is executed and creates boot images, software
3 packages, and node information, and wherein said master node places the boot images and
4 software packages in said software package storage means and the node information in said
5 node information storage means.

1 31. A computer-readable medium as recited in Claim 21, wherein said retrieving step
2 creates said node's preferred software version information from said node information
3 storage means based on functional features requested by said node.

1 32. An apparatus of software loading and initialization in a distributed network of nodes,
2 comprising:
3 a master node;
4 software package storage means on said master node for storing software packages
5 and boot images that the nodes in the network will be using as well as older
6 versions that are kept for regressing a node back to a previous boot image or
7 software package version;

8 node information storage means on said master node for storing preferred software
9 version information, node type, and other pertinent information for each node
10 in the network;
11 wherein a node performing an initial boot requests a boot image and software
12 packages from said master node;
13 means for retrieving said node's preferred software version information from said
14 node information storage means;
15 means for extracting boot image and software packages from said software package
16 storage means using said node's preferred software version information; and
17 means for delivering the extracted boot image and software packages to said node.

1 33. An apparatus as recited in Claim 32, wherein said node stores the extracted boot
2 image and software packages in its local persistent storage and wherein software version
3 information is extracted from the software packages and stored in the local persistent storage.

1 34. An apparatus as recited in Claim 33, wherein said node, based on a command from
2 said master node, does not store the software packages in the local persistent storage device,
3 allowing said master node to download test software packages to said node and temporarily
4 run said node using the test software packages, and wherein when said node reboots, the test
5 software packages will no longer exist on said node.

1 35. An apparatus as recited in Claim 33, wherein said node reboots and executes the boot
2 image stored in the local persistent storage, and wherein said node verifies the software
3 version information with said master node.

1 36. An apparatus as recited in Claim 35, wherein if said node has the correct software
2 versions, then said node completes booting by executing the software packages stored in the
3 local persistent storage.

1 37. An apparatus as recited in Claim 35, wherein if said node does not have the correct
2 software versions, said master node retrieves correct software packages from said software
3 package storage means and sends the correct software packages to said node, and wherein
4 said node stores the correct software packages in the local persistent storage and completes
5 booting by executing the correct software packages stored in the local persistent storage.

1 38. An apparatus as recited in Claim 29, wherein the master node has the ability to
2 categorize nodes into classes where all of the nodes in a particular class of nodes have the
3 same software configuration and may have differing processor types.

1 39. An apparatus as recited in Claim 32, wherein a software package contains version
2 information, dependency information, and other metadata information pertaining to software
3 in the package.

1 40. An apparatus as recited in Claim 32, wherein a boot image is customized for a
2 particular type of node and provides basic low-level communications.

1 41. An apparatus as recited in Claim 32, wherein a user installs a composite image onto
2 said master node which is executed and creates boot images, software packages, and node
3 information, and wherein said master node places the boot images and software packages in
4 said software package storage means and the node information in said node information
5 storage means.

1 42. An apparatus as recited in Claim 32, wherein said retrieving means creates said
2 node's preferred software version information from said node information storage means
3 based on functional features requested by said node.